E 14

```
PPPPPPPPPPPP        AAAAAAAAA     TTTTTTTTTTTTTTT     CCCCCCCCCCCC  HHH         HHH
PPPPPPPPPPPP        AAAAAAAAA     TTTTTTTTTTTTTTT     CCCCCCCCCCCC  HHH         HHH
PPPPPPPPPPPP        AAAAAAAAA     TTTTTTTTTTTTTTT     CCCCCCCCCCCC  HHH         HHH
PPP         PPP  AAA         AAA        TTT        CCC              HHH         HHH
PPP         PPP  AAA         AAA        TTT        CCC              HHH         HHH
PPP         PPP  AAA         AAA        TTT        CCC              HHH         HHH
PPP         PPP  AAA         AAA        TTT        CCC              HHH         HHH
PPP         PPP  AAA         AAA        TTT        CCC              HHH         HHH
PPPPPPPPPPPP     AAA         AAA        TTT        CCC              HHHHHHHHHHHHHHH
PPPPPPPPPPPP     AAA         AAA        TTT        CCC              HHHHHHHHHHHHHHH
PPPPPPPPPPPP     AAA         AAA        TTT        CCC              HHHHHHHHHHHHHHH
PPP              AAAAAAAAAAAAAAA        TTT        CCC              HHH         HHH
PPP              AAAAAAAAAAAAAAA        TTT        CCC              HHH         HHH
PPP              AAAAAAAAAAAAAAA        TTT        CCC              HHH         HHH
PPP              AAA         AAA        TTT        CCC              HHH         HHH
PPP              AAA         AAA        TTT        CCC              HHH         HHH
PPP              AAA         AAA        TTT        CCC              HHH         HHH
PPP              AAA         AAA        TTT        CCC              HHH         HHH
PPP              AAA         AAA        TTT           CCCCCCCCCCCC  HHH         HHH
PPP              AAA         AAA        TTT           CCCCCCCCCCCC  HHH         HHH
PPP              AAA         AAA        TTT           CCCCCCCCCCCC  HHH         HHH
```

```
                                        I 9
**FILE**ID**PATCON




PPPPPPPP      AAAAAA    TTTTTTTTTT    CCCCCCCC    OOOOOO    NN      NN
PPPPPPPP      AAAAAA    TTTTTTTTTT    CCCCCCCC    OOOOOO    NN      NN
PP      PP  AA      AA      TT      CC          OO      OO  NN      NN
PP      PP  AA      AA      TT      CC          OO      OO  NN      NN
PP      PP  AA      AA      TT      CC          OO      OO  NNNN    NN
PP      PP  AA      AA      TT      CC          OO      OO  NNNN    NN
PPPPPPPP    AA      AA      TT      CC          OO      OO  NN  NN  NN
PPPPPPPP    AA      AA      TT      CC          OO      OO  NN  NN  NN
PP          AAAAAAAAAA      TT      CC          OO      OO  NN    NNNN
PP          AAAAAAAAAA      TT      CC          OO      OO  NN    NNNN
PP          AA      AA      TT      CC          OO      OO  NN      NN      ....
PP          AA      AA      TT      CC          OO      OO  NN      NN      ....
PP          AA      AA      TT        CCCCCCCC    OOOOOO    NN      NN      ....
PP          AA      AA      TT        CCCCCCCC    OOOOOO    NN      NN      ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
  0 MODULE PATCON (
L 0                     %IF %VARIANT EQL 1
  0                     %THEN
  0                             ADDRESSING_MODE (EXTERNAL = LONG_RELATIVE, NONEXTERNAL = LONG_RELATIVE),
  0                     %FI
  0                     IDENT = 'V04-000') =
  1 BEGIN
  1
  1 !****************************************************************************
  1 !*                                                                          *
  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
  1 !*  ALL RIGHTS RESERVED.                                                    *
  1 !*                                                                          *
  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
  1 !*  TRANSFERRED.                                                            *
  1 !*                                                                          *
  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
  1 !*  CORPORATION.                                                            *
  1 !*                                                                          *
  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
  1 !*                                                                          *
  1 !*                                                                          *
  1 !****************************************************************************
  1
  1 ! FACILITY:     PATCH
  1 !
  1 !++
  1 ! FUNCTIONAL DESCRIPTION:
  1 !
  1 !       CONVERSION ROUTINES.
  1 !
  1 ! Version:      V02-010
  1 !
  1 ! History:
  1 !       Author:
  1 !               Carol Peters, 18 May 1976: Version 01
  1 !
  1 ! Modified by:
  1 !
  1 !       V02-010 PCG0001         Peter George            02-FEB-1981
  1 !               Add require statement for LIB$:PATDEF.REQ
  1 !
  1 !       01.09   CNH0013         Chris Hume              27-Aug-1979     13:30
  1 !               Added double byte OPcode support.  Changed use of PAT$CONV_R_50
  1 !               to the RTL routine R50ASC.  Removed the former from this module.
  1 !
  1 ! MODIFICATIONS:
  1 !
  1 !  NO   DATE            PROGRAMMER              PURPOSE
  1 !  --   ----            ----------              -------
```

```
1 !
1 !     00      19-OCT-77       K.D. MORSE              CONVERT VERSION 7 FOR PATCH.
1 !     01      4-JAN-78        K.D. MORSE              NO CHANGES FOR VERS 8.
1 !     02      21-FEB-78       K.D. MORSE              USE EMUL FOR OVERFLOW CHECK.
1 !     03      24-MAR-78       K.D. MORSE              REPLACE SELECT WITH IF...THEN
1 !                                                     AS IN DEBUG OVERFLOW CHECK AS
1 !                                                     THIS SAVES BYTES. (9)
1 !     04      04-APR-78       K.D. MORSE              NO CHANGES FOR 10.
1 !     05      25-APR-78       K.D. MORSE              CONVERT TO NATIVE COMPILER.
1 !     06      18-MAY-78       K.D. MORSE              NO CHANGES FOR VERS 11.
1 !     07      13-JUN-78       K.D. MORSE              ADD FAO COUNTS TO SIGNALS.
1 !     08      19-JUN-78       K.D. MORSE              NO CHANGES FOR VERS 12-13.
1 !
1 !--
```

```
1 FORWARD ROUTINE
1           PAT$RADX_CONVRT : NOVALUE;                                 ! Converts a string to a value
1
1 LIBRARY 'SYS$LIBRARY:STARLET.L32';
1 REQUIRE 'SRC$:VXSMAC.REQ';
1 REQUIRE 'SRC$:PATPCT.REQ';
1 REQUIRE 'SRC$:PATGEN.REQ';
1 REQUIRE 'LIB$:PATDEF.REQ';                                 ! Defines literals
1 REQUIRE 'LIB$:PATMSG.REQ';
1
1 EXTERNAL
1           PAT$GB_MOD_PTR: REF VECTOR [, BYTE];                       ! Pointer to current modes
```

```
1 GLOBAL ROUTINE PAT$RADX_CONVRT (STRING_ADDR, VALUE_ADDR) : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       Converts an ascii string to a longword value in the current radix.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       PAT$RADX_CONVRT ()
1 !
1 ! INPUTS:
1 !
1 !       STRING_ADDR     - Address of ascii string
1 !       VALUE_ADDR      - Address in which to put converted value
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       Contents of PAT$gb_mod_ptr [mode_radix], which is the current
1 !       radix.
1 !
1 ! OUTPUTS:
1 !
1 !       none
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       A signal and unwind occurs if the conversion fails.
1 !       The converted value is placed in the address passed as the
1 !       second argument.
1 !
1 ! ROUTINE value:
1 !
1 !       novalue
1 !
1 ! SIDE EFFECTS:
1 !
1 !       none
1 !
1 !--
1
2 BEGIN
2
2 BUILTIN
2       EMUL;                                           ! Longword mul and add to get quadword
2
2 MAP
2       STRING_ADDR: REF VECTOR [, BYTE],
2       VALUE_ADDR: REF VECTOR;
2
2 LOCAL
2       GIVE_MESSAGE,                                   ! Error flag
2       value : VECTOR[2,LONG],                         ! Converted value
2       NEGATE,                                         ! Negative number flag
2       CHAR;                                           ! Character-holding variable
2
2 VALUE[0] = 0;
```

```
;  143          0691  2 VALUE[1] = 0;
;  144          0692  2 GIVE_MESSAGE = FALSE;
;  145          0693  2 NEGATE = FALSE;
;  146          0694  2 INCR N FROM 0 TO (NUM_MAX_LENGTH - 1) DO
;  147          0695  3          BEGIN
;  148          0696  3          IF (CHAR = .STRING_ADDR [.N]) EQL 0 THEN EXITLOOP;
;  149          0697  4          IF (.N EQL 0)
;  150          0698  3          THEN
;  151          0699  4                   BEGIN
;  152          0700  5                   IF (.CHAR EQL '-')
;  153          0701  4                   THEN
;  154          0702  5                            BEGIN
;  155          0703  5                            NEGATE = TRUE;
;  156          0704  5                            CHAR = '0'
;  157          0705  5                            END
;  158          0706  4                   ELSE
;  159          0707  5                            IF (.CHAR EQL '+')
;  160          0708  4                            THEN CHAR = '0';
;  161          0709  3                   END;
;  162          0710  4          IF ((.CHAR GEQ '0') AND (.CHAR LEQ '9'))
;  163          0711  3          THEN
;  164          0712  3                   CHAR = .CHAR - '0'
;  165          0713  3          ELSE
;  166          0714  4                   IF ((.CHAR GEQ 'A') AND (.CHAR LEQ 'F'))
;  167          0715  3                   THEN
;  168          0716  3                            CHAR = .CHAR - 'A' + 10
;  169          0717  3                   ELSE
;  170          0718  3                            CHAR = 256;
;  171          0719  3          IF .CHAR GEQ .PAT$GB_MOD_PTR [MODE_RADIX]
;  172          0720  3          THEN
;  173          0721  3                   GIVE_MESSAGE = PAT$_INVNUMBER
;  174          0722  3
;  175          0723  3          ELSE
;  176          0724  3                   EMUL(VALUE[0], %REF(.PAT$GB_MOD_PTR[MODE_RADIX]), CHAR, VALUE);
;  177          0725  3          IF .VALUE[1] NEQ 0
;  178          0726  3          THEN
;  179          0727  3                   GIVE_MESSAGE = PAT$_NUMTRUNC;                      ! Numeric overflow
;  180          0728  2          END;
;  181          0729  3 IF (.GIVE_MESSAGE NEQ 0)
;  182          0730  2 THEN
;  183          0731  2          SIGNAL(.GIVE_MESSAGE);
;  184          0732  2 IF .NEGATE
;  185          0733  2 THEN
;  186          0734  2          VALUE[0] = - .VALUE[0];
;  187          0735  2 VALUE_ADDR [0] = .VALUE[0];
;  188          0736  1 END;


                                                          .TITLE   PATCON
                                                          .IDENT   \V04-000\

                                                          .EXTRN   PAT$GB_MOD_PTR

                                                          .PSECT   _PAT$CODE,NOWRT,2

                                    001C 00000            .ENTRY   PAT$RADX_CONVRT, Save R2,R3,R4              ; 0634
                 5E              04  C2 00002             SUBL2    #4, SP                                      ;
```

```
                                          7E  D4 00005          CLRL    VALUE                                       ; 0690
                              04          AE  D4 00007          CLRL    VALUE+4                                     ; 0691
                                          53  7C 0000A          CLRQ    GIVE_MESSAGE                                ; 0692
                                          52  D4 0000C          CLRL    N                                           ; 0719
                    50        04 BC42     9A 0000E 1$:          MOVZBL  @STRING_ADDR[N], CHAR                       ; 0696
                                          71  13 00013          BEQL    11$                                         ;
                                          52  D5 00015          TSTL    N                                           ; 0697
                                          12  12 00017          BNEQ    4$                                          ;
                    2D                    50  D1 00019          CMPL    CHAR, #45                                   ; 0700
                                          05  12 0001C          BNEQ    2$                                          ;
                    54                    01  D0 0001E          MOVL    #1, NEGATE                                  ; 0703
                                          05  11 00021          BRB     3$                                          ; 0704
                    2B                    50  D1 00023 2$:      CMPL    CHAR, #43                                   ; 0707
                                          03  12 00026          BNEQ    4$                                          ;
                    50                    30  D0 00028 3$:      MOVL    #48, CHAR                                   ; 0708
                    30                    50  D1 0002B 4$:      CMPL    CHAR, #48                                   ; 0710
                                          0A  19 0002E          BLSS    5$                                          ;
                    39                    50  D1 00030          CMPL    CHAR, #57                                   ;
                                          05  14 00033          BGTR    5$                                          ;
                    50                    30  C2 00035          SUBL2   #48, CHAR                                   ; 0712
                                          1C  11 00038          BRB     7$                                          ;
           00000041 8F                    50  D1 0003A 5$:      CMPL    CHAR, #65                                   ; 0714
                                          0E  19 00041          BLSS    6$                                          ;
           00000046 8F                    50  D1 00043          CMPL    CHAR, #70                                   ;
                                          05  14 0004A          BGTR    6$                                          ;
                    50                    37  C2 0004C          SUBL2   #55, CHAR                                   ; 0716
                                          05  11 0004F          BRB     7$                                          ;
                    50        0100        8F  3C 00051 6$:      MOVZWL  #256, CHAR                                  ; 0718
50 00000000G  FF    08                    00  ED 00056 7$:      CMPZV   #0, #8, @PAT$GB_MOD_PTR, CHAR               ; 0719
                                          09  14 0005F          BGTR    8$                                          ;
                    53 006D80EA           8F  D0 00061          MOVL    #7176426, GIVE_MESSAGE                      ; 0721
                                          0C  11 00068          BRB     9$                                          ;
                    51 00000000G          FF  9A 0006A 8$:      MOVZBL  @PAT$GB_MOD_PTR, R1                         ; 0724
      6E            50                    51  6E  7A 00071      EMUL    VALUE, R1, CHAR, VALUE                      ;
                              04          AE  D5 00076 9$:      TSTL    VALUE+4                                     ; 0725
                                          07  13 00079          BEQL    10$                                         ;
                    53 006D8023           8F  D0 0007B          MOVL    #7176227, GIVE_MESSAGE                      ; 0727
              88    52                    13  F3 00082 10$:     AOBLEQ  #19, N, 1$                                  ; 0694
                                          53  D5 00086 11$:     TSTL    GIVE_MESSAGE                                ; 0729
                                          09  13 00088          BEQL    12$                                         ;
                                          53  DD 0008A          PUSHL   GIVE_MESSAGE                                ; 0731
           00000000G 00                   01  FB 0008C          CALLS   #1, LIB$SIGNAL                              ;
                    03                    54  E9 00093 12$:     BLBC    NEGATE, 13$                                 ; 0732
                    6E                    6E  CE 00096          MNEGL   VALUE, VALUE                                ; 0734
                 08 BC                    6E  D0 00099 13$:     MOVL    VALUE, @VALUE_ADDR                          ; 0735
                                              04 0009D          RET                                                 ; 0736

; Routine Size:  158 bytes,    Routine Base:  _PAT$CODE + 0000
```

```
                                                                  C 10
PATCON                                                            16-Sep-1984 00:26:45    VAX-11 Bliss-32 V4.0-742              Page   7
V04-000                                                           14-Sep-1984 12:52:30    DISK$VMSMASTER:[PATCH.SRC]PATCON.B32;1   (4)

;  190          0737  1 END
;  191          0738  0 ELUDOM



                                                                          .EXTRN  LIB$SIGNAL

;                                         PSECT SUMMARY
;
;       Name                          Bytes                        Attributes
;
;   _PAT$CODE                              158  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)



;                                 Library Statistics
;
;                                              -------- Symbols --------        Pages        Processing
;       File                                   Total    Loaded   Percent       Mapped       Time
;
;   _$255$DUA28:[SYSLIB]STARLET.L32;1           9776         4         0          581         00:01.0




;                                          COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/VARIANT:1/LIS=LIS$:PATCON/OBJ=OBJ$:PATCON MSRC$:PATCON/UPDATE=(ENH$:PATCON)

; Size:          158 code + 0 data bytes
; Run Time:         00:11.5
; Elapsed Time:     00:49.4
; Lines/CPU Min:     3857
; Lexemes/CPU-Min: 55426
; Memory Used:  107 pages
; Compilation Complete
```

0300 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

PATARI
LIS

PATCMD
LIS

PATECO
LIS

PATCON
LIS

PATENC
LIS

PATBAS
LIS

PATBLD
LIS